

## Практическое занятие № 4

### Обобщенные табличные выражения

**Цель занятия.** Получить практические навыки использования обобщенных табличных выражений

Обобщенные табличные выражения (Common Table Expression - CTE) позволяют в более удобной форме составлять сложные запросы. CTE можно рассматривать как определение виртуальных таблиц, которые существуют только для текущего запроса. CTE записываются в предложении WITH, и их можно рассматривать как представление или виртуальную таблицу внутри запроса. Преимуществом WITH является возможность многократного обращения к колонкам подзапроса в других частях запроса.

Запрос с использованием предложения WITH может формировать несколько временных таблиц и записывается по следующим правилам:

```
WITH имя_таблицы_1 as (подзапрос_1),  
     имя_таблицы_2 as (подзапрос_2),  
     ... //после последнего подзапроса запятая не ставится  
SELECT ... // основной запрос
```

При этом каждый следующий подзапрос может обращаться к колонкам любого из предшествующих. Основной запрос может использовать как основные таблицы БД, так и виртуальные таблицы, подготовленные предложением WITH.

Рассмотрим примеры использования WITH на примере схемы о сотрудниках организации.

Организация включает несколько отделов. В штате организации состоят сотрудники. Каждый *Сотрудник* организации работает в некотором отделе (задается колонкой внешнего ключа - Код отдела). В каждом *Отделе* имеется начальник (задается внешним ключом Код начальника - ссылка на сотрудника, который является начальником). Логическая и физическая модели данных представлены соответственно на рис.1 и рис.2.

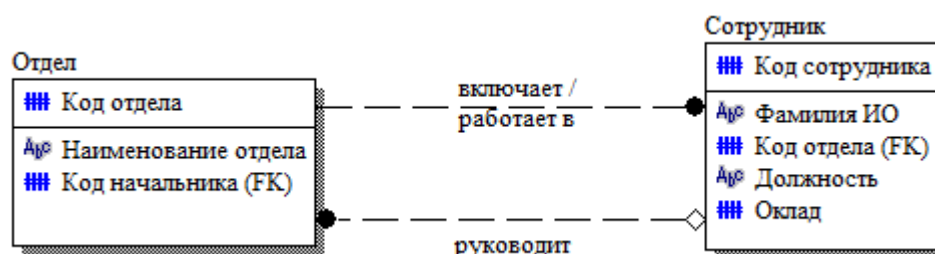


Рис.1 Логическая модель

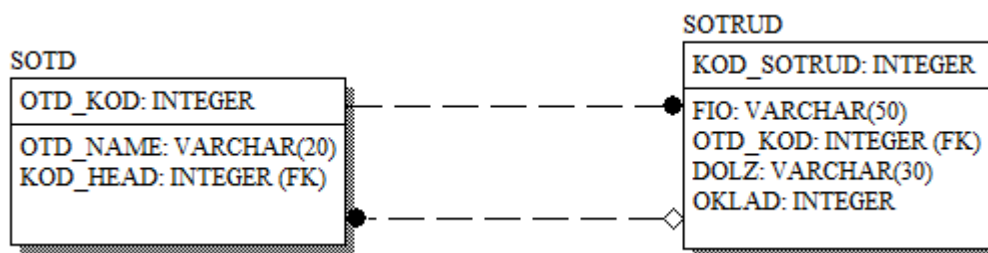


Рис.2 Физическая модель

Выполним запросы на следующих данных.

### Отдел (SOTD)

Otd_Kod	Otd_Name	Kod_Head
1	Техотдел	1
2	Отдел продаж	6
3	Отдел закупок	10

### Сотрудник (SOTRUD)

Sotrud_Kod	FIO	Otd_Kod	Dolz	Oklad
1	Марков В.В.	1	Нач. отдела	150000
2	Семенов В.А.	1	Инженер	50000
3	Исаев А.М.	2	Менеджер	40000
4	Акимова А.С.	3	Ст.инженер	70000
5	Павленко И.И.	1	Вед.инженер	95000
6	Морозова А.И.	2	Нач. отдела	140000
7	Сомова Е.Н.	2	Вед.менеджер	65000
8	Пименов С.А.	1	Инженер	45000
9	Селезнева Н.М.	2	Менеджер	55000
10	Конева А.М.	3	Нач.отдела	130000

Для тестирования примеров создадим 2 виртуальные таблицы с использованием предложения WITH. Это удобно для тестирования, т.к. не требует создавать таблицы в БД.

```

WITH Sotd (otd_kod,otd_name,head_kod) as (
    SELECT 1,'Техотдел',1
    union SELECT 2,'Отдел продаж',6
    union SELECT 3,'Отдел закупок',10),
Sotrud (sotrud_kod,fio,otd_kod,dolz,oklad) as (
    SELECT 1,'Марков В.В.',1,'Нач.отдела',150000
    union SELECT 2,'Семенов В.А.',1,'Инженер',50000
    union SELECT 3,'Исаев А.М.',2,'Менеджер',40000
    union SELECT 4,'Акимова А.С.',3,'Ст.инженер',70000
  
```

```

union SELECT 5, 'Павленко И.И.', 1, 'Вед.инженер', 95000
union SELECT 6, 'Морозова А.И.', 2, 'Нач.отдела', 140000
union SELECT 7, 'Сомова Е.Н.', 2, 'Вед.менеджер', 65000
union SELECT 8, 'Пименов С.А.', 1, 'Инженер', 45000
union SELECT 9, 'Селезнева Н.М.', 2, 'Менеджер', 55000
union SELECT 10, 'Конева А.М.', 3, 'Нач.отдела', 130000)

```

*Пример 1.* Вывести информацию о сотрудниках и численности отдела, в которых они работают.

Вначале создадим оператор запроса с использованием обычного подзапроса. Одну из таблиц в разделе FROM сформируем подзапросом, который возвращает численность всех отделов:

```

SELECT FIO, so.otd_sotr
FROM   Sotrud s,
      (SELECT otd_kod, COUNT(*) AS kol_sotr // таблица-
        FROM sotrud                      // подзапрос
        GROUP BY otd_kod) so
WHERE  s.otd_kod = so.otd_kod;

```

*Замечание.* Т.к. таблиц Отдел и Сотрудник в БД мы не создавали, то для получения результата запроса создадим виртуальные таблицы и заполним их данными, как было описано выше с помощью предложения WITH перед основным запросом:

```

WITH sotd (otd_kod, otd_name, head_kod) as (
    SELECT 1...), // формирование таблицы Отдел с данными
sotrud (kod_sotrud, fio, otd_kod, dolz, oklad) as (
    SELECT 1,...), //формирование таблицы Сотрудник с
данными

```

Результат запроса

	FIO	kol_sotr
1	Акимова А.С.	2
2	Конева А.М.	2
3	Исаев А.М.	4
4	Морозова А.И.	4
5	Сомова Е.Н.	4
6	Селезнева Н.М.	4
7	Семенов В.А.	4
8	Павленко И.И.	4
9	Пименов С.А.	4
10	Марков В.В.	4

Тот же запрос можно записать с использованием обобщенного табличного выражения. Подзапрос, формирующий вспомогательную таблицу (в примере он определяет численность каждого отдела), перенесем в предложение WITH, а затем используем в основном операторе запроса. Основной запрос обращается к колонкам как базовой таблицы sotrud, так и временной otd\_count

```
WITH sotd (otd_kod,otd_name,head_kod) as (
    SELECT 1...), // формирование таблицы Отдел с данными
sotrud (kod_sotrud,fio,otd_kod,dolz,oklad) as (
    SELECT 1,...), //формирование таблицы Сотрудник с данными
// временная таблица, определяющая численность отделов
otd_count AS (SELECT otd_kod, COUNT(*) AS kol_sotr
    FROM sotrud
    GROUP BY otd_kod) //
// Основной запрос
SELECT FIO,oc.kol_sotr
FROM sotrud as s,
    otd_count as oc
WHERE s.otd_kod = oc.otd_kod;
```

Результат будет таким же, как и при использовании обычного подзапроса.

*Пример 2.* Вывести фамилии сотрудников отделов, фамилии руководителей и количество сотрудников в отделе.

Также сначала составим оператора запроса с подзапросом.

```
SELECT s1.FIO,
    otd_count.kol_sotr,
    s2.FIO AS HEAD_FIO
FROM sotrud s1,
    (SELECT otd_kod, COUNT(*) AS kol_sotr
    FROM sotrud
    GROUP BY otd_kod) otd_count, //подзапрос
    sotd,
    sotrud s2
WHERE s1.otd_kod = otd_count.otd_kod
AND s1.otd_kod = sotd.otd_kod
AND sotd.head_kod = s2.sotrud_kod;
```

Подзапрос-таблица otd\_count используется для получения численности отдела, в котором работает сотрудник. Таблица sotd используется для получения кода сотрудника, который является начальником

отдела. Копия таблицы *Сотрудник* s2 используется, чтобы получить фамилию сотрудника-начальника отдела.

Результат запроса:

	FIO	kol_sotr	HEAD_FIO
1	Акимова А.С.		2 Конева А.М.
2	Конева А.М.		2 Конева А.М.
3	Исаев А.М.		4 Морозова А.И.
4	Морозова А.И.		4 Морозова А.И.
5	Сомова Е.Н.		4 Морозова А.И.
6	Селезнева Н.М.		4 Морозова А.И.
7	Семенов В.А.		4 Марков В.В.
8	Павленко И.И.		4 Марков В.В.
9	Пименов С.А.		4 Марков В.В.
10	Марков В.В.		4 Марков В.В.

Перепишем запрос с использованием предложения WITH.

```
WITH // Создание базовых таблиц
  sotrud (otd_kod,otd_name,head_kod) as (
    SELECT 1...),
  sotrud (kod_sotrud,fio,otd_kod,dolz,oklad) as (
    SELECT 1,...),
// Создание временных таблиц
Otd_count AS (
  SELECT otd_kod, COUNT(*) AS kol_sotr
  FROM   sotrud
  GROUP BY otd_kod)
SELECT s1.FIO,
       s1.kol_sotr,
       s2.FIO AS HEAD_FIO
FROM   sotrud s1,
       otd_count s1,
       sotrud s2
WHERE  s1.otd_kod = s1.otd_kod
AND    s1.otd_kod =sotrud.otd_kod
AND    sotrud.head_kod = s2.sotrud_kod;
```

Результат будет таким же.

**Пример 3** Вывести список отделов в которых средний оклад превышает средний оклад в организации

```
WITH // Создание базовых таблиц .....
      // Создание временных таблиц
```

```

otd_oklads AS (
    SELECT otd_name,
           AVG(OKLAD) as avg_oklad
    FROM   sotrud s, soto so
    WHERE  s.otd_kod = so.otd_kod
    GROUP BY otd_name),
avg_oklad AS (
    SELECT AVG(avg_oklad) as avg_org FROM otd_oklads)
SELECT OTD_NAME, round(AVG_OKLAD,2)
FROM   otd_oklads, avg_oklad
WHERE  avg_oklad > avg_org
ORDER BY otd_name;

```

Первая временная таблица с именем **otd\_oklads** определяет средние оклады по отделам

Вторая временная таблица с именем **avg\_oklad** на основе данных первого подзапроса определяет средний оклад по организации.

Результирующий запрос выводит отделы и средние оклады в них, если этот оклад превышает средний оклад по организации. При этом условие соединения двух виртуальных таблиц (**otd\_oklads**, **avg\_oklad**) не задается, т.к. вторая таблица содержит одну строку. Функция **round** округляет результат до двух дробных разрядов.

Результат.

Results	
OTD_NAME	AVG_OKLAD
1 Отдел закупок	100,000.00

**Упражнение 1.** Вывести список сотрудников (отдел, ФИО, должность, оклад), у которых оклад ниже среднего оклада по организации. Данные упорядочить по отделам, ФИО

**Упражнение 2.** Вывести список должностей (должность, средний оклад), у которых средний оклад меньше среднего оклада по организации. Данные упорядочить в порядке убывания окладов

### Задание

1. Выполните упражнения на учебной БД
2. Составьте похожие запросы для вашей БД